

*Exam 1.p.fa.2024*  
*Principles of Computing*  
*30<sup>th</sup> of October, 2024*



**1 Answer the following questions either True or False.**

I. The *root* of a UNIX-based file system is a directory named `/root`.

- ☐ True  
☐ False

II. The `cd` command can be used to change a directory's name.

- ☐ True  
☐ False

III. The name `~` is an alias for the current user's "*home*" directory.

- ☐ True  
☐ False

IV. `mv(pumpkins.py, ~/patch)` will move `pumpkins.py` to the `patch` directory in the user's home folder.

- ☐ True  
☐ False

V. `touch closet/skeletons.csv` will create a new file named `skeletons.csv` within `closet` in the home folder.

- ☐ True  
☐ False

VI. `", ".join(["this is Halloween", "this is Halloween"])[19::1]` is valid syntax.

- ☐ True  
☐ False

VII. Every value in *Python* is an object, and every object has a type.

- ☐ True  
☐ False

VIII. Every method is a function in *Python*.

- ☐ True  
☐ False

IX. Every function returns a value.

- ☐ True  
☐ False

X. `print("Hello, world!")` returns an object of type `str`.

- ☐ True  
☐ False

The first five questions on this page are about the *shell*; the rest are about *Python*.

**2 Answer the following questions without justification.**

i. What is the value of `scary_string` at the end of the following block?

```
1 scary_string = "i am the one hiding under your bed".split(" ")
2 scary_string[0:2] += ["not"]
3 scary_string = " ".join(scary_string)
```

.....

ii. What is the result of the following expression?

```
1 len("".join("everyone hail to the pumpkin song\n".split(" ")))
```

.....

iii. What is the value of `spooky_func(" ")` with the definition below?

```
1 def spooky_func(ghost):
2     x = "i am the shadow on the moon at night".split(ghost)
3     return x.insert(2, "not")
```

.....

iv. How many times will line 4 be evaluated in the block of code below?

```
1 def rattle(something):
2     return len(something) % 2 == 0
3     bones = list(range(101))
4     while rattle(bones):
5         bones = bones[:bones[-1]/2]
```

.....

v. What value is returned as a result of calling the function below?

```
1 def trick_or_treat():
2     ll = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
3     fear = 0
4     for x in ll:
5         for y, z in enumerate(x):
6             if z % (y + 1) == 0:
7                 fear += z
8             if fear == 13:
9                 return "trick"
10    return "treat"
```

.....

### 3 Answer the following questions without justification.

- i. What *type of error* will be thrown when this code is run?<sup>1</sup>

```

1 def jump_scare(days_left):
2     return days_left == 7
3 upper_bound = input("Enter your maximum tolerance for terror: ")
4 for i in range(upper_bound):
5     print(jump_scare(days_left))

```

<sup>1</sup> You may assume the user provides valid, sensible input (e.g., 10) when prompted.

- ii. Indicate *all lines* in the following code block that would *throw an error*.

```

1 dungeon = []
2 for i in range(5):
3     room = []
4     for j in range(5):
5         room += [i*j]
6     dungeon.append(room)
7 if sum(dungeon) % 13 == 0:
8     return "creepy"
9 return "wet"

```

- iii. The block of code below throws an error; on *what line* does it occur?

```

1 from random import randint
2 weight = "0"
3 giles_corey = "185"
4 while plea not in ["guilty", "not guilty"]:
5     weight += int(f"{randint(1, 10)}")
6     giles_corey -= int(weight)
7     plea = "more weight"

```

- iv. What is `monster_sort([3, 1, -1, 0, 5, 3])` based on the code below?

```

1 def monster_sort(ll):
2     rr = []
3     for l in ll:
4         scared = False
5         for i, r in enumerate(rr):
6             if (l < r) and (not scared):
7                 rr.insert(i, l)
8                 scared = True
9         if not scared:
10             rr.append(l)
11     return rr

```

#### 4 Program a solution to the following question in Python.

Given two vectors  $x = [x_0, x_1, \dots, x_{n-1}]$  and  $y = [y_0, y_1, \dots, y_{n-1}]$  of the same positive integer length  $n$ , we compute the *inner product* of  $x$  with  $y$ , which we denote  $\langle x, y \rangle$ , as follows.

$$\langle x, y \rangle = x_0 y_0 + x_1 y_1 + \dots + x_{n-1} y_{n-1}$$

For example, the inner product of  $[1, 2, 3]$  and  $[2, 4, 8]$  is below.

$$\begin{aligned}\langle [1, 2, 3], [2, 4, 8] \rangle &= 1 \cdot 2 + 2 \cdot 4 + 3 \cdot 8 \\ &= 2 + 8 + 24 \\ &= 34\end{aligned}$$

Write a function in *Python* that takes two lists of floating point numbers representing *vectors* as inputs and returns their *inner product* as output *if they are the same length*, but returns the string `"ERROR"` otherwise.

**RESTRICTIONS:**

- No use of `import` statements.
- No use of methods for any type.
- No use of the built-in functions:
  - `sorted()`
  - `sum()`
  - `max()`
  - `min()`
  - `abs()`
- No use of functions, methods, types, or control structures that we have not yet covered in class nor problem sets.

**NOTE:** you may name your function and your variables whatever you'd like.